Overview

- Types of numerical approximation and their coding
- Array operations

- *Numerical methods* constructs approximation sequences $\{y_n\}_{n\in\mathbb{N}}$ to converge to the solution of a mathematical problem $y = P(x)$, $\lim_{n\to\infty} y_n = y$.
- *Numerical analysis* studies the convergence behavior: conditions for convergence, order and rate of convergence.
- *Computation theory* constructs algorithms within some model of computation and studies the behavior of these algorithms, in particular the computational effort required for each step of the algorithm

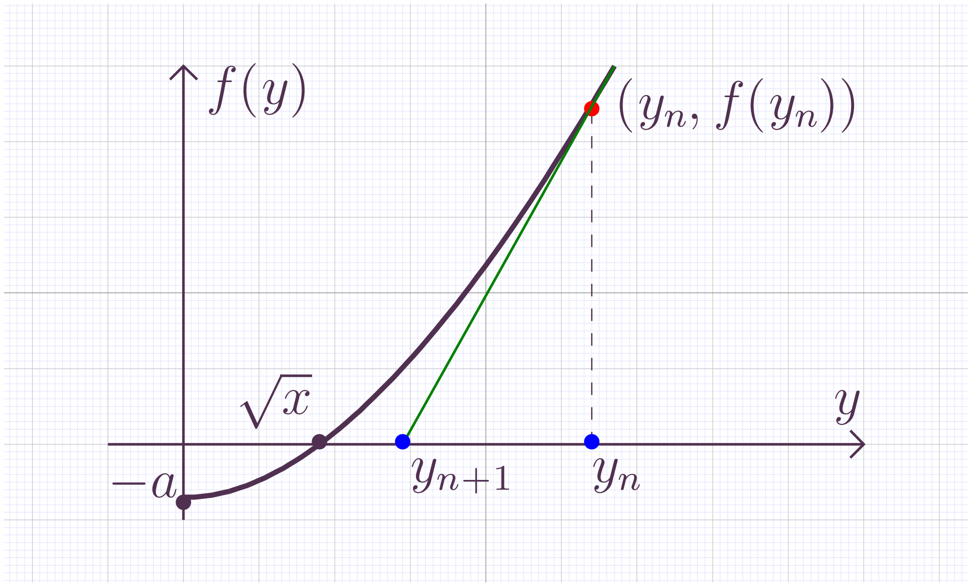
$$y_1 = 1,\ y_{n+1} = \frac{1}{2}\left(y_n + \frac{x}{y_n}\right),$$

$$\lim_{n\to\infty} y_n = \sqrt{x} = P(x).$$

$$f(y) = R(x, y) = x - y^2,\ r_n = R(x, y_n)$$

$$L_n(y) = f'(y_n)(y - y_n) + f(y_n)$$

$$L_n(y_{n+1}) = 0 \Rightarrow y_{n+1} = \frac{1}{2}\left(y_n + \frac{a}{y_n}\right)$$



$$y_1 = 1,\ y_{n+1} = \frac{1}{2}\left(y_n + \frac{a}{y_n}\right)$$

Suppose $y_n = \sqrt{x}(1+\delta_n)$, signifying that y_n is "δ_n-close" to $\sqrt{x}$. Then, from $y_{n+1} - y_n = \frac{1}{2}(x/y_n - y_n)$ obtain

$$y_{n+1} - y_n = \frac{1}{2}\left[\frac{x}{\sqrt{x}(1+\delta_n)} - \sqrt{x}(1+\delta_n)\right] = \frac{\sqrt{x}}{2}\left[\frac{1}{1+\delta_n} - 1 - \delta_n\right] \Rightarrow$$

$$y_{n+1} - y_n = \frac{\sqrt{x}}{2}[(1 - \delta_n + \delta_n^2 - \delta_n^3 + \cdots) - 1 - \delta_n] = \delta_n\sqrt{x}\,\frac{2-\delta_n}{2(1-\delta_n)} \leqslant \delta_n\sqrt{x}.$$

For $\forall\varepsilon > 0$ choose N_ε such that $|\delta_{N_\varepsilon}| < \varepsilon/\sqrt{x}$ in which case $|y_{n+1} - y_n| < \varepsilon$, hence $\{y_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathbb{R}$, and converges.

- Newton's method for square root extraction: $f(y) = a - y^2 = 0$

$$y_1 = 1,\, y_{n+1} = \frac{1}{2}\left(y_n + \frac{a}{y_n}\right)$$

  - At each iteration carry out 1 addition and two multiplications. Computational effort is $\mathcal{O}(2n)$ FLOPs (floating point operations)

- Newton's method for finding zeros of f, $f(y) = 0$

$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}.$$

  - At each iteration carry out 1 evaluation of f and 1 evaluation of f'. Computational effort is $\mathcal{O}(n)$ f-evals, and $\mathcal{O}(n)$ f'-evals.

- Additive corrections

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots,$$

$$L_n = \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}, L_n = L_{n-1} + \frac{(-1)^n}{2n+1} \to \frac{\pi}{4}.$$

- Multiplicative corrections

$$S_n = \left(\frac{2}{1} \cdot \frac{2}{3}\right) \cdot \left(\frac{4}{3} \cdot \frac{4}{5}\right) \cdot \left(\frac{6}{5} \cdot \frac{6}{7}\right) \ldots,$$

$$S_n = \prod_{k=1}^{n} \frac{4k^2}{4k^2-1} = S_{n-1} \cdot \frac{4n^2}{4n^2-1}, S_n \to \frac{\pi}{2}.$$

- Continued fractions

$$\pi + 3 = 6 + \cfrac{1^2}{6 + \cfrac{3^2}{6 + \cfrac{5^2}{6 + \ddots}}},$$

$$F_n = b_0 + \operatorname*{K}_{k=1}^{n} \frac{a_k}{b_k} = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots}}},$$

$$y_n = 0,\ y_{n-1} = \frac{a_{n-1}}{b_{n-1} + y_n},\ F_n = b_0 + y_1.$$

Definition. *The sum of matrices* $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$

$$\boldsymbol{A} = [a_{i,j}], \boldsymbol{B} = [b_{i,j}]$$

is the matrix $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$ *with components*

$$\boldsymbol{C} = [c_{i,j}], c_{i,j} = a_{i,j} + b_{i,j}$$

Definition. *The scalar multiplication of matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *by* $\alpha \in \mathbb{R}$ *is* $\boldsymbol{B} = \alpha \boldsymbol{A}$

$$\boldsymbol{A} = [a_{i,j}], \boldsymbol{B} = [b_{i,j}] = [\alpha a_{i,j}]$$

Definition. *Consider matrices* $\boldsymbol{A} = [\,\boldsymbol{a}_1 \;\; \ldots \;\; \boldsymbol{a}_n\,] \in \mathbb{R}^{m \times n}$, *and* $\boldsymbol{X} = [\,\boldsymbol{x}_1 \;\; \ldots \;\; \boldsymbol{x}_p\,] \in \mathbb{R}^{n \times p}$. *The* matrix product $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X}$ *is a matrix* $\boldsymbol{B} = [\,\boldsymbol{b}_1 \;\; \ldots \;\; \boldsymbol{b}_p\,] \in \mathbb{R}^{m \times p}$ *with column vectors given by the matrix vector products*

$$\boldsymbol{b}_k = \boldsymbol{A}\boldsymbol{x}_k, \text{for } k = 1, 2 \ldots, p.$$

- A matrix-matrix product is simply a set of matrix-vector products, and hence expresses multiple linear combinations in a concise way.
- The dimensions of the matrices must be compatible, the number of rows of $\boldsymbol{X}$ must equal the number of columns of $\boldsymbol{A}$.
- A matrix-vector product is a special case of a matrix-matrix product when $p = 1$.
- We often write $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X}$ in terms of columns as

$$[\,\boldsymbol{b}_1 \;\; \ldots \;\; \boldsymbol{b}_p\,] = \boldsymbol{A}[\,\boldsymbol{x}_1 \;\; \ldots \;\; \boldsymbol{x}_p\,] = [\,\boldsymbol{A}\boldsymbol{x}_1 \;\; \ldots \;\; \boldsymbol{A}\boldsymbol{x}_p\,]$$

Definition. *Consider matrices* $\boldsymbol{A} = [\ a_{i,j}\ ] \in \mathbb{R}^{m\times n}$, *and* $\boldsymbol{X} = [x_{i,j}] \in \mathbb{R}^{n\times p}$. *The matrix product* $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X} = [b_{i,j}]$ *is a matrix* $\boldsymbol{B} \in \mathbb{R}^{m\times p}$ *with components*

$$b_{i,j} = a_{i,1}\,x_{1,j} + a_{i,2}\,x_{2,j} + \cdots + a_{i,n}\,x_{n,j} = \sum_{k=1}^{n} a_{i,k}\,b_{k,j}$$

$$\boldsymbol{A}\boldsymbol{X} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & & a_{m,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & & x_{m,n} \end{bmatrix}$$

$$b_{2,1} = a_{2,1}\,x_{1,1} + a_{2,2}\,x_{2,1} + \cdots + a_{2,n}\,x_{n,1}$$

- $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains n column vectors with m components each

$$\boldsymbol{A} = [\ \boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \dots \quad \boldsymbol{a}_n\ ]$$

- The transpose switches rows and columns

$$\boldsymbol{A}^T = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix} \in \mathbb{R}^{n \times m}$$

has n rows and m columns

- Matrix block addition

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{B} \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{F} & \boldsymbol{E} \end{bmatrix},$$

$$\boldsymbol{A} + \boldsymbol{D} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{B} \end{bmatrix} + \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{F} & \boldsymbol{E} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}+\boldsymbol{E} & \boldsymbol{C}+\boldsymbol{F} \\ \boldsymbol{C}+\boldsymbol{F} & \boldsymbol{B}+\boldsymbol{E} \end{bmatrix}$$

- Matrix block multiplication

$$\boldsymbol{A}\boldsymbol{D} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{F} & \boldsymbol{E} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}\boldsymbol{E}+\boldsymbol{C}\boldsymbol{F} & \boldsymbol{B}\boldsymbol{F}+\boldsymbol{C}\boldsymbol{E} \\ \boldsymbol{C}\boldsymbol{E}+\boldsymbol{B}\boldsymbol{F} & \boldsymbol{C}\boldsymbol{F}+\boldsymbol{B}\boldsymbol{E} \end{bmatrix}$$

- Matrix block transposition

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{V} \\ \boldsymbol{X} & \boldsymbol{Y} \end{bmatrix}, \boldsymbol{M}^T = \begin{bmatrix} \boldsymbol{U}^T & \boldsymbol{X}^T \\ \boldsymbol{V}^T & \boldsymbol{Y}^T \end{bmatrix}$$